AMENDMENTS TO THE SPECIFICATION

Please amend the specification as follows:

Amend the paragraph on page 7, lines 10-21, as follows:

Figure 3 shows a load queue 30, having four entries in this simplified illustration, and a store queue 50, also having four entries or locations (numbered at 38) in this simplified illustration. The issue queue 10 sends load instructions (i.e., the instructions in locations 1, 4, 5, 7, and 8 of box 20 in Figure 2) to the load queue 30 in a random order that depends on the availability of the resources needed to execute the instruction, such as having the indicated memory location free. The load queue 30 places each received load instruction in a particular one of the four possible locations; that is the load instructions are preassigned a correct location in the load queue 30 that keeps the proper program order even though the load instructions are sent to the load queue 30 from the issue queue 10 (box 20) in random order. This is accomplished by a modulus operation performed (at 104) on the order of the load operation instruction in the program order, for example, on the LSN.

Amend the paragraph starting on page 8, line 22 through page 9, line 1, as follows:

When the particular operation represented by the entry LD 0 is completed and the data is used by the program, then the entry 0 in the load queue 30 should be cleared for the next instruction, in the current illustrative example the LD 4 instruction. This is done by switching the valid bits 36 from the valid (i.e., the instruction currently in the queue entry is still being worked on and is not yet complete) state of zero to an invalid state (i.e., the instruction has already been sent and is no longer valid). The MSB bit remains set at zero until the first of the next series of load instructions (i.e., the LD 4 instruction in the present illustrative example) has issued. As each individual queue entry is freed up by the completion of its stored instruction, the MSB bit value is switched (at 100) in a process known as "being complemented." This prepares the queue entry location to accept information regarding the next instruction that may legitimately occupy that particular queue location. Thus, the issue queue 10 knows whether or not the load queue 30 (or the store queue 50, etc.) is ready to accept more randomly sent load instructions, (i.e., the queue 30, 50 is no longer full) by the combination of the valid bit value 36 and the MSB bit value 32.

Insert the following paragraph on page 9, at line 3:

The instruction to be executed is compared to its respective memory queue entry, and hence the MSB bit of the issuing instruction is compared against the MSB value stored in that queue entry. If the MSB values are the same in the issuing instruction and queue entry, and the queue entry is currently in an INVALID state, then that instruction may be allowed to write information into that queue entry location. On the other hand, if the MSB values differ between the issuing instruction and the queue entry location (compare operation at 102), or if the queue entry is currently in a VALID state, then the instruction will not be allowed to write information into that queue entry location, and a queue full flag is set, which lets the main processor know to stop issuing instructions.